PACIFIC STATES MARINE FISHERIES COMMISSION
205 SE Spokane Street, SUITE 100, Portland, OREGON 97202
PHONE (503) 595-3100 FAX (503) 595-3232

2023-2024 Pacific Islands Logit Integration RFP
Questions & Answers
08/14/2023

1. What is your anticipated budget for this work?

   A:

   1. The budget for development and support of the Catchit and Logit System is not to exceed $300,000
   2. The budget for development, testing, and documentation of the Developer Tooling and Application Framework is not to exceed $400,000

2. Do you have a required or anticipated timeline for this work?

   A:

   1. The goal for completion of the Catchit and Logit System is approximately one year of agile development.
   2. The goal for completion of the Developer Tooling and Application Framework is approximately six months of agile development.

3. Will you consider extending the due date to accommodate Q&A?

   A: Not at this point in the process.

4. Will you please provide a list of existing bugs and feature requests for the Catchit and Logit applications?

   A: No. We have a list of approximately 50 bugs/feature requests for Catchit. Only bugs that are considered major will be included in the integration. Others will be targeted after the integration is complete.

5. Will you please provide data model diagrams for the Catchit and Logit databases?

   A: See Appendix A and B.

6. Will you please send us any system architecture diagrams that you have for Catchit and Logit that illustrates the existing cloud infrastructure for the systems?

   A: We do not have full documentation or diagrams currently. Both applications are deployed to AWS using various AWS technologies. Deployment configurations are managed through Terraform.

7. What kind of authentication schemes must the scaffolding project support?

A: The expectation is that the tooling is developed to be extensible to allow for different authentication strategies to be selected. We are looking for vendor expertise on implementing at least one common authentication strategy, such as OAuth2.

8. Are there any specific technology requirements for the tooling framework?

   A: The tooling can be based on existing libraries if it helps expedite the process, but we will need documentation on how the PSMFC/NOAA tooling will be kept up to date. Since the preferred target is a React.js boilerplate it would seem node.js would be the underlying technology.

9. Between the Developer Tooling and the Catchit and Logit Systems, is there urgency or a need to prioritize one project over the other?

   A: While the RFP targets both efforts to be completed to be considered successful, there is interest in having the tooling completed quicker to be made available to use. If staffing provides, then the project tasks can be worked in parallel, otherwise the Developer toolkit should be completed first.

10. Is there an existing methodology (Scrum/Kanban) and systems the vendor must work within for managing engineering scope?

    A: We currently follow a hybrid approach with bi-weekly scums, sprint planning, and we are using JIRA kanban to track task progress. See Appendix C.

11. What technologies are the current applications and APIs written in?

    A:

Catchit	Dashboard	React 16 (JavaScript)
	Admin Dashboard	React 16 (JavaScript)
	Fishing Report PWA	React 16 (JavaScript)
	Vendor Report PWA	React 16 (JavaScript)
	Report Server	Express 4 (JavaScript) Knex ORM
	Admin Server	Express 4 (JavaScript) Forest Admin
Logit	Data Entry App	React 18 (TypeScript)
	API Server	Express 4 (TypeScript) Prisma ORM

12. What are the current database technologies in use?

    A: Both systems use MySQL 8, Redis

13. Is there anything that could cause the data structure to change during the course of development?

A: Yes. One key aspect of the integration is to move the tables from Catchit into Logit. This will require analysis of where common data can be merged into single tables and where whole tables can move without modification.

14. Are there any restrictions for technologies to meet the requirements of this contract?

    A: No proprietary technologies. We will accept recommendations from the vendor, but PSMFC/NOAA maintain the right to final approval. The current applications utilize JavaScript/TypeScript/Node.js technologies. Since this is not a rewrite, but rather an integration, the expectation is that JavaScript (related) technologies will be used.

15. What are the existing and/or approved cloud providers for application deployment?

    A: Both Catchit and Logit are deployed to Amazon Web Services.

16. Are there any existing use cases or user stories that could be provided regarding the Developer Tooling and Application Framework scope.

    A: We are looking for the vendor to help groom the user stories to focus on priority tasks. Also, see RFP 2.9. and Appendix C

17. Are there any crucial dates the Contractor should be aware of regarding the cutover of data?

    A: A specific date has not been identified. We are targeting having the integration complete within a year.

18. What is the current authentication schemes for both applications?

    A: Both applications utilize Passport.js library using the passport-local and passport-jwt strategies.

19. What is the preferred unified authentication scheme?

    A: We are looking for help to find the best solution. We have different users who present challenges when trying to find a single solution. There are trained users who can work with standard login credentials and are able to navigate through self service password reset. We also have less technical users that will often be accessing the PWA application in an offline manner, and may need more assistance when gaining access.

20. Are there any data security or compliance requirements the vendor should know about?

    A: For the Developer Tooling there are several considerations that need to be made.

    1. Must meet PSMFC/NOAA security standards and encryption standards.
    2. ADA Section 508 compliant.
    3. WCAG 2.0 AA compliant.

    We have provided this link https://engineering.18f.gov/frontend/ to give some considerations on the standards that are being targeted. See Appendix C.

21. Will the scaffolding be a client/server web application, front-end application, mobile application or PWA?

    A: The scaffolding will be for the client application. It should be able to scaffold either standard React.js web application or a PWA application, based on options passed. We know that there will be aspects of the application that rely on server side components. For example, authentication and error handling. While the scaffolding will not need to generate any server side code, we will need clear documentation showing examples of what needs to be implemented server side, possibly in multiple target technologies such as node.js, .Net Core, and PHP CodeIgniter.

22. Has the PSMFC/NOAA branding and style guide already been created or is it a requirement of contract fulfillment?

    A: We expect the scaffolding to provide optional component and style libraries. As a default we will be targeting USWDS 3.0 https://designsystem.digital.gov/ which has an available component library by TrussWorks Inc. https://github.com/trussworks/react-uswds

23. In section 2.7, what is defined as a "host type"?

    A: The Developer tooling is for scaffolding a web based client. As covered in response 21. We will need code examples at a minimum and working boilerplate as a best case, for developing the API component that handles the Authentication and Authorization requests. Working with the vendor we can determine the initial host types that will need to be provided.

24. Is there a central directory of existing users the application will need to authenticate with using SSO?

    A: No

25. In item 2.8, are test requirements unit, integration and system testing individually for the target coverage or a combination of the various types?

    A: The reference in 2.8 is specifically referring to unit tests. The overall understanding we want to convey is that the Developer Tooling is intended to be used across many departments and projects. As such, we need to make sure that it is well tested and documented to facilitate the widest adoption possible.

26. In item 2.8, is there a list of devices and operating systems test coverage should support?

    A: There is not an existing list of required devices or operating systems. The goal is to provide a React application platform that is either a conventional web application or a Progressive Web Application (PWA), that may have a target audience that is internal or public. We understand that it covers basically all combinations of devices and host systems. Initially we think the core requirements for the non-PWA app is, responsive web design that works with Google Chrome, running on Windows 10 or Mac OS. For the PWA we think targeting Android OS and iOS with Google Chrome Mobile as the target browser. We will be looking to the vendor to provide expertise on how to address this issue.

27. Are there any existing test systems or technologies in place the vendor must integrate with?

    A: The Logit application has an existing CI/CD pipeline defined using Github actions.

28. Is there a target date for the mentioned production release?

    A:

    1. For the Catchit/Logit integration, a specific date has not been identified. We are targeting having the integration complete within a year, which would include releasing updated

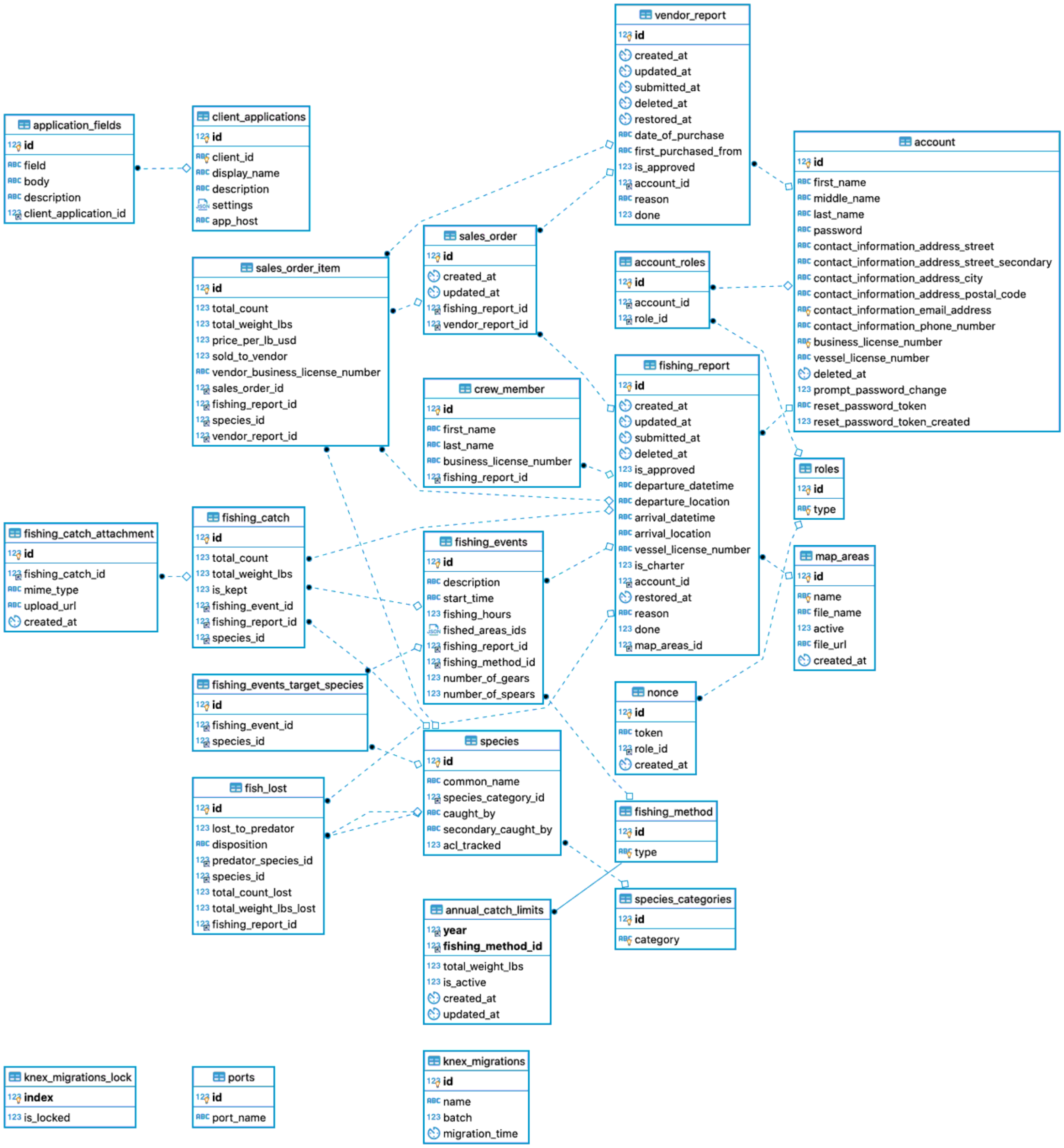
versions of the production client PWA applications that communicate with the integrated backend. If vendor staffing cannot support working both tasks simultaneously, we expect the Developer Tooling to be completed first. In this case, the integration target date would move back accordingly.

2. For the Developer Tooling and Application Framework we are targeting having a working MVP within six months.

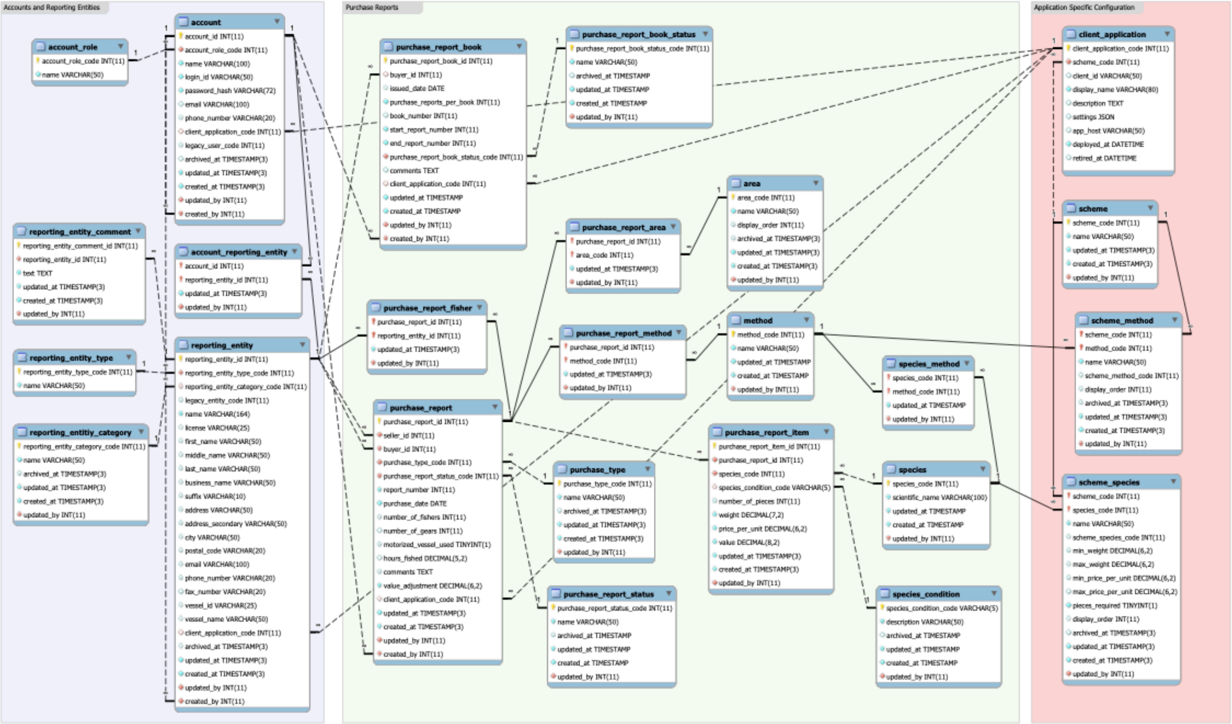
29. Is there an existing SLA or business continuity requirements to meet for unanticipated outages or bug fixes?

    A: No

Appendix A: Catchit ERD



Appendix B: Logit ERD



Appendix C: Quality Assurance Surveillance Plan (QASP)

The deliverables of this project are fully-tested and fully-documented functioning software at the end of each sprint (sprint length TBD). The following chart sets forth the performance standards and quality levels the deliverables provided by the Contractor must meet, and the methods PSMFC/NOAA will use to assess the standard and quality levels of those deliverables.

Deliverable	Performance Standard	Acceptable Quality Level	Method of Assessment
1. Timely Delivery	Every sprint cycle (TBD) fully-tested and fully-documented functioning software is deployed into the PSMFC/NOAA GitHub repository.	Functioning software does not have to be a completed feature, but needs to be usable by PSMFC/NOAA for testing and to inform further development.	Does PSMFC/NOAA receive Functioning software for every open epic in each sprint?
2. Regular Backlog Prioritization	Regular backlog refinement and prioritization meetings between the Contractor and PSMFC/NOAA.	A minimum backlog refinement and prioritization meeting (TBD) to ensure the user stories are completed in order of priority.	Are development efforts and deliverables representative of backlog refinements and prioritization?
3. Testing Coverage	Functioning software delivered must have testing coverage no lower than 90%.	Minimum of 90% test coverage of all code, and all major components are meaningfully tested and passed before delivered to PSMFC/NOAA.	Are there a combination of manual review and automated tests documented and/or provided to PSMFC/NOAA for every open epic in each sprint to ensure appropriate test coverage?
4. Properly Styled Code	The Contractor must ensure that code is consistently styled and documented throughout the framework and does not contradict best practices for open source technologies used in the stack.	Functioning software has 0 linting errors and 0 warnings, or any linting errors and warnings are otherwise documented by the Contractor to be resolved.	Are there a combination of manual review and automated tests documented and/or provided to PSMFC/NOAA for every open epic in each sprint to ensure consistent styling?
5. Accessibility Standards	The Contractor must ensure that the design system and styling used in this framework meet WCAG 2.0 standards and standards for accessibility per Section 508 of the Rehabilitation Act.	Code bootstrapped with this framework pass A and AA benchmarks from automated accessibility checker, E.g., Siteimprove and or tools such as Lighthouse and Pa11y.	Are there a combination of manual review and automated tests documented and/or provided to PSMFC/NOAA for every open epic in each sprint to ensure accessibility standards are being met?

6. Deployment	Every sprint cycle (TBD) fully-tested and fully-documented functioning software is committed into the NFMS GitHub repository and deployable on PSMFC/NOAA development environments.	The functioning software can be cloned from the PSMFC/NOAA repository and deployed on PSMFC/NOAA development environments in one or a few documented build commands, e.g., `npx create-nmfs-app`	Does the functioning software deploy from PSMFC/NOAA GitHub to local development environments using the build commands documented by the Contractor?
7. Documentation	Every sprint cycle (TBD) must include updated comprehensive and developer-oriented documentation including a list of dependencies.	Comprehensive and developer-oriented documentation are provided, via GitHub, for all major components of the framework including all dependencies.	Are comprehensive and developer-oriented documentation provided for every open epic in each sprint?
8. Security	The Contractor must ensure that functioning software meets PSMFC/NOAA security standards and encryption standards.	Code bootstrapped with this framework pass all tests from automated scans, E.g., Qualys.	Are there a combination of manual review and automated tests documented and/or provided to PSMFC/NOAA for every open epic in each sprint to ensure security standards are being met?